

## AMENDMENTS TO THE CLAIMS

1. (Currently Amended) A method for processing objects at a data processing system in a network, the method comprising processing logic used for:  
receiving a message at a computing device; ~~and~~  
determining that a message header in the message indicates that the message relates to a fragment; ~~and determining that the message header in the message indicates that the fragment is cacheable[[:]], wherein the message header comprises an indication that the fragment is non-cacheable to~~ no-cache directive data for not caching the fragment for non-fragment-supporting cache management units and an indication that the fragment is cacheable to caching directive data for caching the fragment for fragment-supporting cache management units;  
determining whether a cache management unit is a non-fragment-supporting cache management unit or a fragment-supporting cache management unit;  
processing the no-cache directive data to generate a no-cache instruction to a non-fragment-supporting cache management unit; and  
processing the caching directive data to generate a no-cache instruction to a non-fragment-supporting cache management unit.
2. (Currently Amended) The method of claim 1 further comprising:  
storing a fragment from the message in a cache maintained by a fragment-supporting cache management unit within the computing device, wherein the fragment-supporting cache management unit operates equivalently in support of fragment caching operations whether the computing device acts as a client, a server, or a hub located throughout the network.
3. (Previously Presented) The method of claim 1 further comprising:  
determining that a message header in the message indicates that a message body portion of the message is a fragment.
- 4-5. (Canceled)

6. (Currently Amended) The method of claim [[5]] 1 wherein the message header comprises an HTTP Cache-Control header ~~with a no-cache directive for non-fragment-supporting cache management units and with a directive for caching the fragment for fragment-supporting cache management units.~~

7-21. (Canceled)

22. (Previously Presented) The method of claim 1 further comprising:  
retrieving a set of dependency identifiers from the message, wherein a dependency identifier is generated by a server that originated the fragment; and storing the set of dependency identifiers in association with a source identifier for the fragment.

23. (Original) The method of claim 22 further comprising:  
receiving an invalidation request message;  
retrieving a dependency identifier from the invalidation request message;  
determining a set of fragments that are associated with the dependency identifier; and  
purging the set of fragments from the cache in response to determining the set of fragments that are associated with the dependency identifier.

24. (Previously Presented) The method of claim 1 further comprising:  
retrieving a set of fragment caching rules from the message, wherein a fragment caching rule determines a manner for generating a cache identifier for the fragment; and  
generating a cache identifier for the fragment in accordance with a fragment caching rule.

25. (Original) The method of claim 24 further comprising:  
uniquely identifying the fragment using the cache identifier.

26 (Original) The method of claim 24 further comprising:  
performing the storing operation using the generated cache identifier for the fragment.

27. (Original) The method of claim 24 further comprising:  
obtaining at least a path portion of a URI (Uniform Resource Identifier) associated with the fragment in order to form a base cache identifier; and

applying a fragment caching rule to the base cache identifier to form a cache identifier for the fragment, wherein a fragment caching rule comprises a set of query parameter names and/or cookie names that are used to obtain name-value pairs that are appended to the base cache identifier.

28-31. (Canceled)

32. (Currently Amended) An apparatus for processing objects at a data processing system in a network, the apparatus comprising:

means for receiving a message at a computing device; ~~and~~

means for determining that a message header in the message indicates that the message relates to a fragment; ~~and determining that the message header in the message indicates that the fragment is cacheable[[:]], wherein the message header comprises an indication that the fragment is non-cacheable to no-cache directive data for not caching the fragment for non-fragment-supporting cache management units and an indication that the fragment is cacheable to caching directive data for caching the fragment for fragment-supporting cache management units;~~

means for determining whether a cache management unit is a non-fragment-supporting cache management unit or a fragment-supporting cache management unit;

means for processing the no-cache directive data to generate a no-cache instruction to a non-fragment-supporting cache management unit; and

means for processing the caching directive data to generate a no-cache instruction to a non-fragment-supporting cache management unit.

33. (Currently Amended) The apparatus of claim 32 further comprising:

means for storing a fragment from the message in a cache maintained by a fragment-supporting cache management unit within the computing device, wherein the fragment-supporting cache management unit operates equivalently in support of fragment caching operations whether the computing device acts as a client, a server, or a hub located throughout the network.

34. (Previously Presented) The apparatus of claim 32 further comprising:  
means for determining that a message header in the message indicates that a message  
body portion of the message is a fragment.

35–36. (Canceled)

37. (Currently Amended) The apparatus of claim 32 wherein the message header  
comprises an HTTP Cache-Control header ~~with a no-cache directive for non-fragment-~~  
~~supporting cache management units and with a directive for caching the fragment for fragment-~~  
~~supporting cache management units.~~

38-52. (Canceled)

53. (Previously Presented) The apparatus of claim 32 further comprising:  
means for retrieving a set of dependency identifiers from the message, wherein a  
dependency identifier is generated by a server that originated the fragment; and  
means for storing the set of dependency identifiers in association with a source identifier  
for the fragment.

54. (Original) The apparatus of claim 53 further comprising:  
means for receiving an invalidation request message; means for retrieving a dependency  
identifier from the invalidation request message;  
means for determining a set of fragments that are associated with the dependency  
identifier; and  
means for purging the set of fragments from the cache in response to determining the set  
of fragments that are associated with the dependency identifier.

55. (Previously Presented) The apparatus of claim 32 further comprising:  
means for retrieving a set of fragment caching rules from the message, wherein a  
fragment caching rule determines a manner for generating a cache identifier for  
the fragment; and  
means for generating a cache identifier for the fragment in accordance with a fragment  
caching rule.

56. (Original) The apparatus of claim 55 further comprising:  
means for uniquely identifying the fragment using the cache identifier.

57. (Original) The apparatus of claim 55 further comprising:  
means for performing the storing operation using the generated cache identifier for the  
fragment.

58. (Original) The apparatus of claim 55 further comprising:  
means for obtaining at least a path portion of a URI (Uniform Resource Identifier)  
associated with the fragment in order to form a base cache identifier; and  
means for applying a fragment caching rule to the base cache identifier to form a cache  
identifier for the fragment, wherein a fragment caching rule comprises a set of  
query parameter names and/or cookie names that are used to obtain name-value  
pairs that are appended to the base cache identifier.

59-62. (Canceled)

63. (Currently Amended) A computer program product in a computer readable  
medium for use in a data processing system in a network for processing objects, the computer  
program product comprising:  
instructions for receiving a message at a computing device; and  
instructions for determining that a message header in the message indicates that the  
message relates to a fragment; ~~and determining that the message header in the~~  
~~message indicates that the fragment is cacheable[[;]], wherein the message header~~  
~~comprises an indication that the fragment is non-cacheable to~~ no-cache directive  
data for not caching the fragment for non-fragment-supporting cache management  
units and an indication that the fragment is cacheable to caching directive data for  
caching the fragment for fragment-supporting cache management units;  
instructions for determining whether a cache management unit is a non-fragment-  
supporting cache management unit or a fragment-supporting cache management  
unit;  
instructions for processing the no-cache directive data to generate a no-cache instruction  
to a non-fragment-supporting cache management unit; and

instructions for processing the caching directive data to generate a no-cache instruction to a non-fragment-supporting cache management unit.

64. (Currently Amended) The computer program product of claim 63 further comprising:

instructions for storing a fragment from the message in a cache maintained by a fragment-supporting cache management unit within the computing device, wherein the fragment-supporting cache management unit operates equivalently in support of fragment caching operations whether the computing device acts as a client, a server, or a hub located throughout the network.

65. (Previously Presented) The computer program product of claim 63 further comprising:

instructions for determining that a message header in the message indicates that a message body portion of the message is a fragment.

66-67. (Canceled)

68. (Currently Amended) The computer program product of claim 63 wherein the message header comprises an HTTP Cache-Control header ~~with a no-cache directive for non-fragment-supporting cache management units and with a directive for caching the fragment for fragment-supporting cache management units.~~

69-83. (Canceled)

84. (Previously Presented) The computer program product of claim 63 further comprising:

instructions for retrieving a set of dependency identifiers from the message, wherein a dependency identifier is generated by a server that originated the fragment; and instructions for storing the set of dependency identifiers in association with a source identifier for the fragment.

85. (Original) The computer program product of claim 84 further comprising: instructions for receiving an invalidation request message;

instructions for retrieving a dependency identifier from the invalidation request message; instructions for determining a set of fragments that are associated with the dependency identifier; and instructions for purging the set of fragments from the cache in response-to determining the set of fragments that are associated with the dependency identifier.

86. (Previously Presented) The computer program product of claim 63 further comprising:

instructions for retrieving a set of fragment caching rules from the message, wherein a fragment caching rule determines a manner for generating a cache identifier for the fragment; and

instructions for generating a cache identifier for the fragment in accordance with a fragment caching rule.

87. (Original) The computer program product of claim 86 further comprising: instructions for uniquely identifying the fragment using the cache identifier.

88. (Original) The computer program product of claim 86 further comprising: instructions for performing the storing operation using the generated cache identifier for the fragment.

89. (Original) The computer program product of claim 86 further comprising: instructions for obtaining at least a path portion of a URI (Uniform Resource Identifier) associated with the fragment in order to form a base cache identifier; and instructions for applying a fragment caching rule to the base cache identifier to form a cache identifier for the fragment, wherein a fragment caching rule comprises a set of query parameter names and/or cookie names that are used to obtain name-value pairs that are appended to the base cache identifier.

90-105. (Canceled)